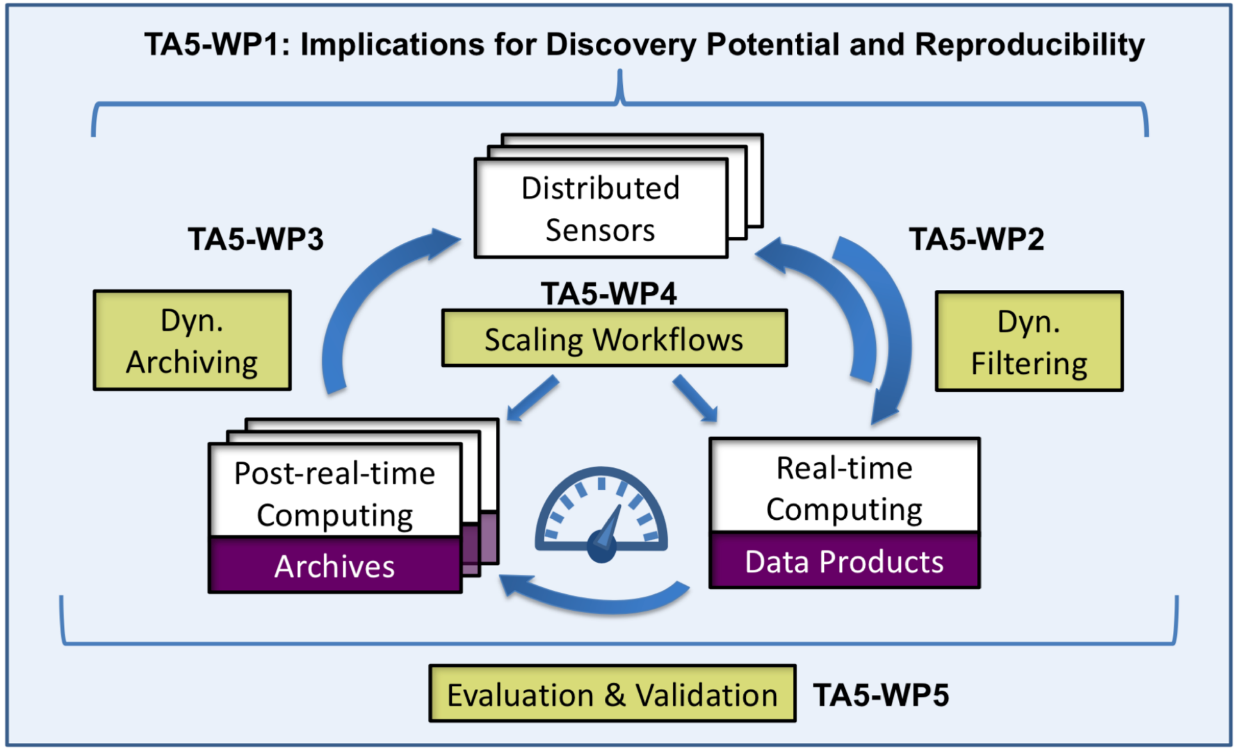
Report from TA5

PUNCH4NFDI Annual Meeting
AIP Potsdam

TA 5: Data Irreversibility

Michael Kramer (MPIfR) and Andreas Redelbach (FIAS)

OVERVIEW OF TA5



Dynamic Life Cycle Model

Focus on progress in WPs

Focus on updates of results

Extra work:
Finalization of deliverables
Publication of results
Deliverable reports

Full list of remaining deliverables in backup

Status of WP2

Dynamic filtering

D-TA5-WP2-3 (MPIfR) Test environment for identifying highly complex (multi-parametric) signals in huge data streams.

→ Links to other deliverables

→ Summary of new developments (next slides)

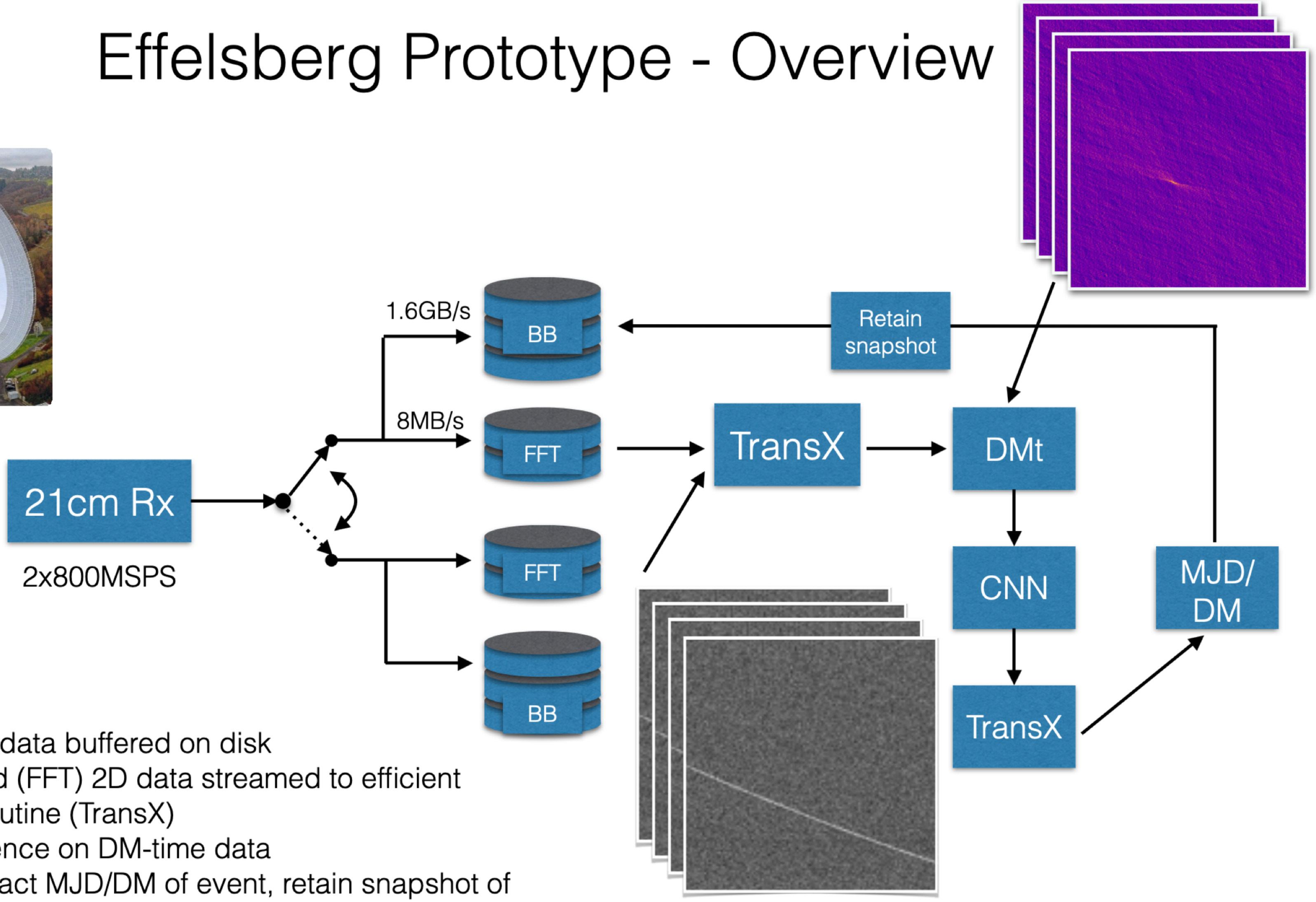
D-TA5-WP2-4 (MPIfR, Dresden, Mainz) Generic tools to both convert trained neural networks into efficient HLS/VHDL FPGA firmware optimised for a real-time, low-latency environment

→ Low-latency requirements fulfilled

→ Effelsberg dataset publicly available (could be linked to DRP)

→ Availability of ATLAS simulation data planned

Status of Effelsberg prototype setup



- High resolution data buffered on disk
- Semi-processed (FFT) 2D data streamed to efficient dedispersion routine (TransX)
- Fast CNN inference on DM-time data
- Wrapper to extract MJD/DM of event, retain snapshot of high resolution data and delete the rest.
- Memory buffers: $\mathcal{O}(\mathrm{L3\ Cache})$

Status of Effelsberg prototype setup

Recent work

- PSRDADA routines to create/manage memory buffers

- TransientX updated with **streaming interfaces**
  - fildb → stream data to memory buffers
  - dbdedispdb → read and dedisperse to buffer

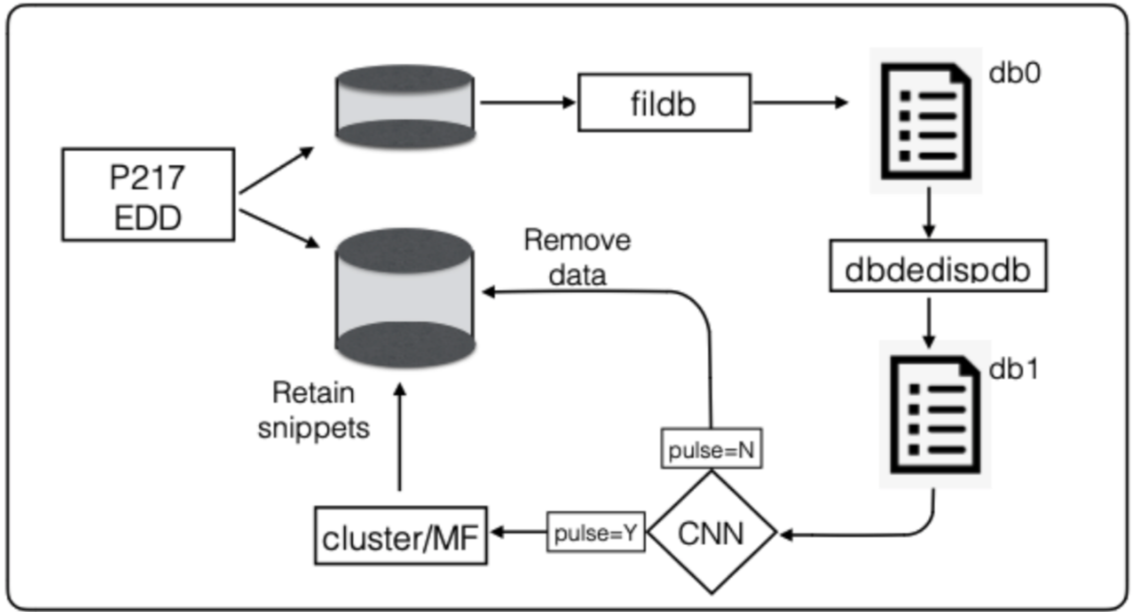
- CNN → ML inference (link below)
  - Development of **minimal models**
  - 800-12000 parameters

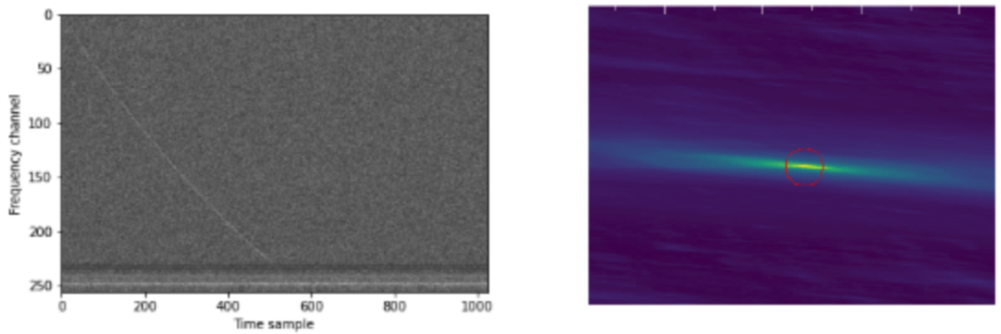
- Ongoing work
  - improve accuracy of minimal ML models
  - Expand sensitivity to wider DM range
  - **integrate ML modules in a parallel Effelsberg pipeline**

https://psrdada.sourceforge.net/download.shtml.html

https://github.com/ypmen/TransientX/tree/dada/src/dada

https://github.com/KazAndr/WP2/tree/developing



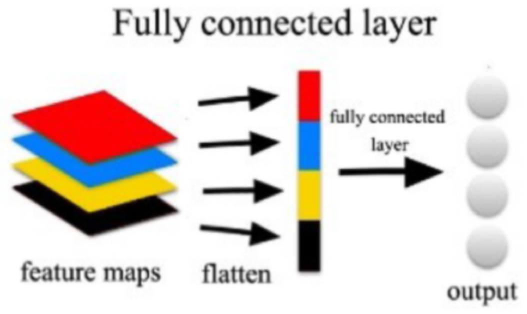


Status of Effelsberg prototype setup

Different models

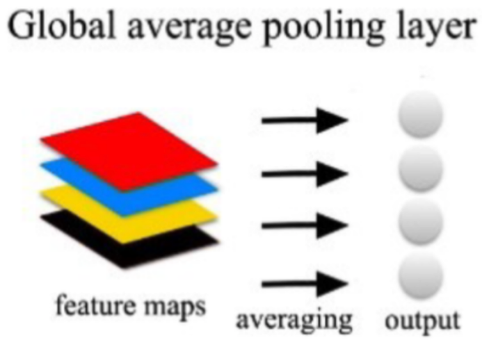
Typical deep models

Architecture	Num. Params
VGG16	≈ 138 M
ResNet50	≈ 25 M
DenseNet121	≈ 8 M
Inception V3	≈ 23 M
Xception	≈ 22 M
Inception-ResNet-V2	≈ 55 M



Minimal models studied

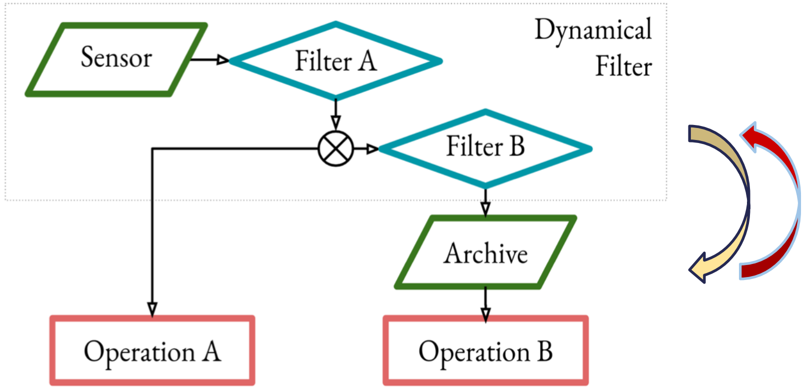
Model name	Resolution	Final size	Parameters	Accuracy (%)
LeNet5 + GAP	128 × 128	182 KB	11 788	99.806
	256 × 256	166 KB	10 408	99.651
MobileNet V1 Trim	128 × 128	98 KB	784	99.653
	256 × 256	100 KB	958	99.522
Mini-SqueezeNet	128 × 128	106 KB	2 044	99.686
	256 × 256	116 KB	2 804	99.547
Shallow CNN	128 × 128	95 KB	4 532	99.767
	256 × 256	75 KB	2 737	99.637



Status of WP3

Dynamic archiving

“A data archive that generates a new dynamic filter.”



Functionalities (concept):

- Estimates information loss
- Quantifies incompleteness
- Estimates cost and time requirements within a workflow
- Converts archive query to real-time filter
- Rapid feedback to online process

D-TA5-WP3-2 Present a framework in which queries to dynamic archives can be transformed into a dynamic filter (as used by some combination of sensors), and vice versa

→ Update by L. Spitler (MPIfR)

→ Application: Re-training minimal CNN models (aka “dynamic filters”) being developed for transient detection in TA5-WP2.

D-TA5-WP3-3 Present methods by which queries to dynamic archives also return an estimate on the potential of information loss, i.e. how well the archive response can be assumed to approximate the response of a real-time sensor

→ Main parts of the work done in the context of AMPEL framework (paper currently written)

→ Coordinated by J. Nordin (Humboldt-Uni)

Dynamic archiving at MPIfR

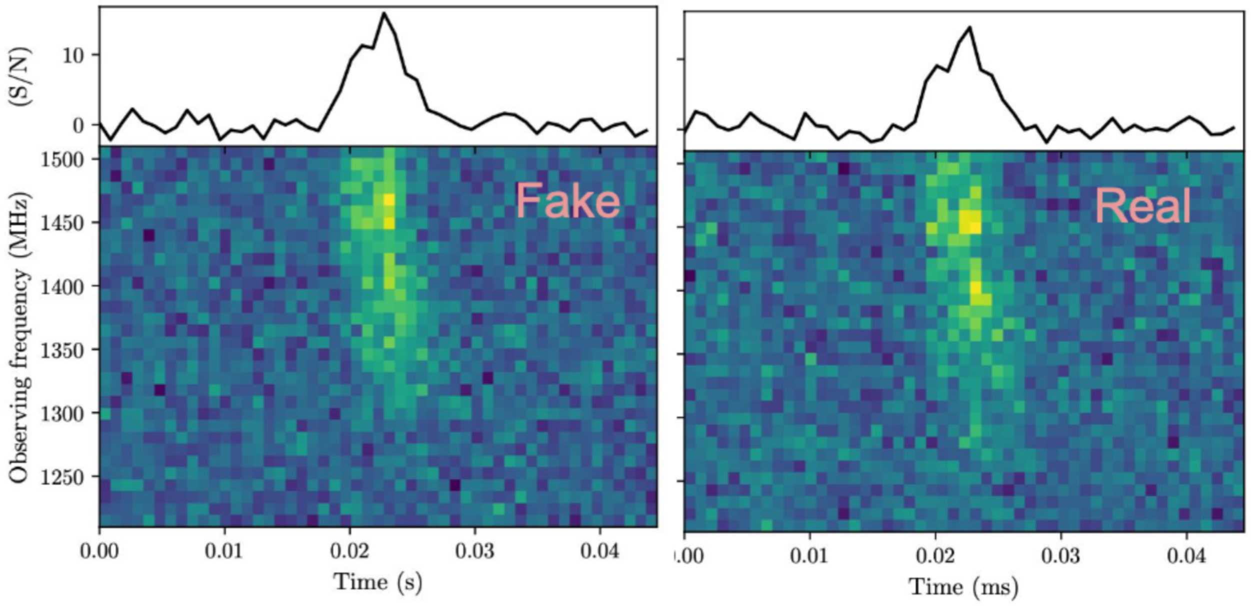
Re-training minimal CNN models

Minimal CNN models may be source-specific and need to be updated regularly.
Re-training of models requires a large (~10^5) and balanced training data set.
This requires generating a synthetic data set by **injecting fake bursts into real data in the archive**.

FRBfaker:
previously developed single pulse injection software able to generate bursts even with complex time-freq. structure

Updates:

- Speed up injection process by inserting multiple bursts in one pass
- Configuration via a json file for ease of reproducibility
- Minimal RFI rejection for more robust calculation of noise statistics



https://gitlab.com/houben.ljm/frb-faker

Houben et al., A&A, submitted, 2025

Dynamic archiving at MPIfR

Next steps

Extract time-frequency snapshots from the data with no (known) astrophysical signal for training

Put the pieces together:

- Generate training data set (dynamic spectrum and DM vs. time images)
- Re-train the minimal CNN model
- Test precision and accuracy to verify that the re-training improved the model performance

Finalization of deliverable **D-TA5-WP3-2**

Status of WP4

Scaling workflows

ML-PPA project:
Development of Machine Learning-based Pipeline for Pulsar Analysis with an overall architecture that addresses real-time and Big Data requirements

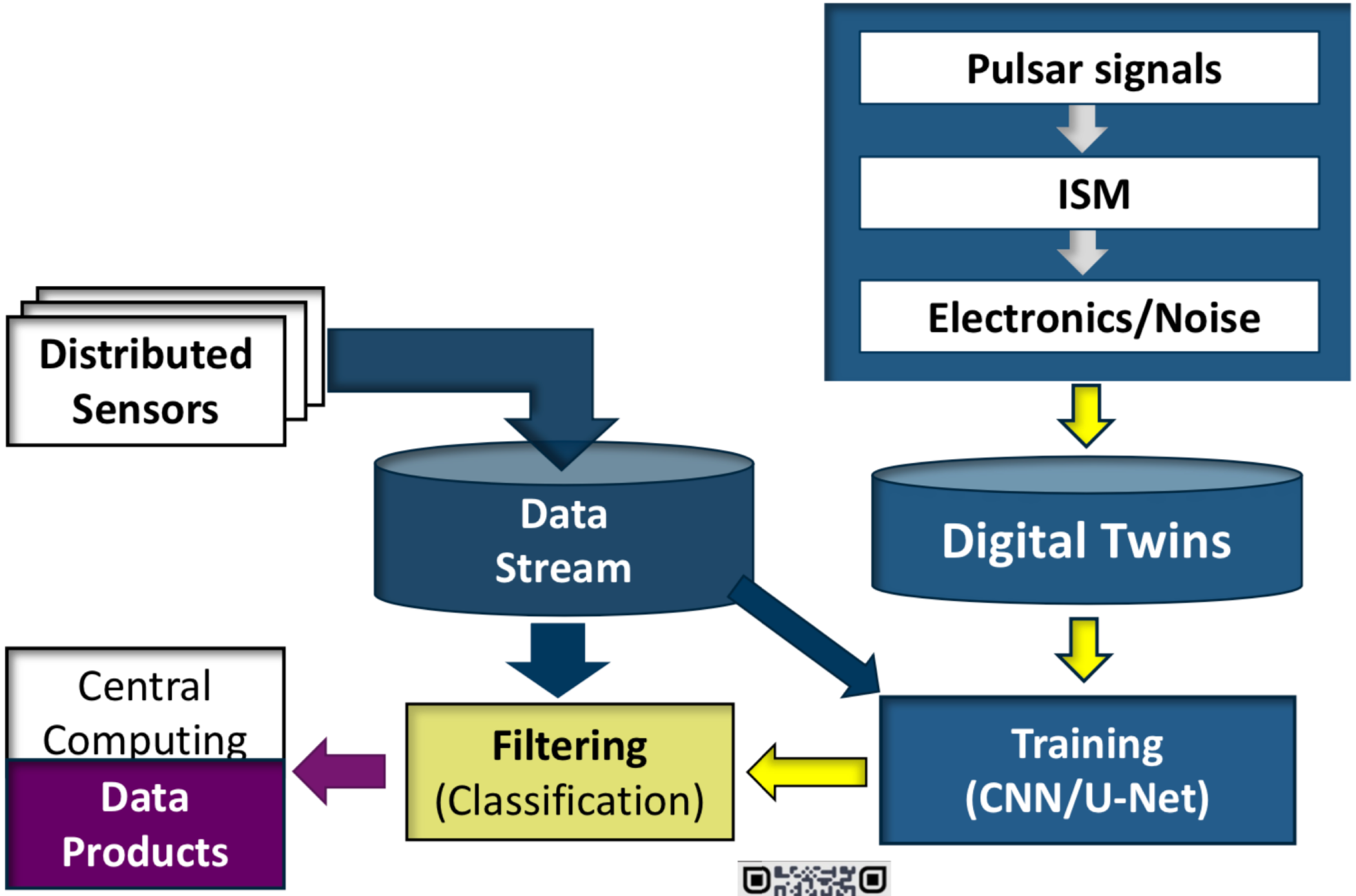
→ Addressing several tasks of WP4
→ https://gitlab.dzastro.de/groups/punch/ml-ppa
→ Update by Hermann (also at ADASS conference)

D-TA5-WP4-3 Caching strategies for processing a set of benchmark files with the evaluated efficiencies and latencies

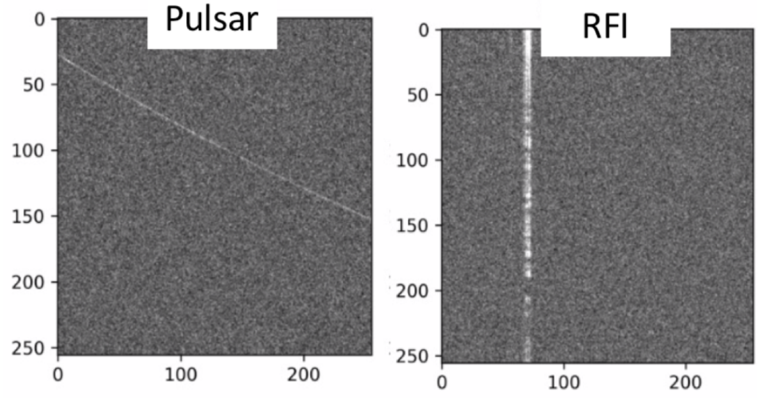
→ Development and setup done, stat. evaluation ongoing, to be finalized in 2025
→ Update by G. Dange (FIAS)

ML-PPA

Machine Learning-based Pipeline for Pulsar Analysis (ML-PPA)

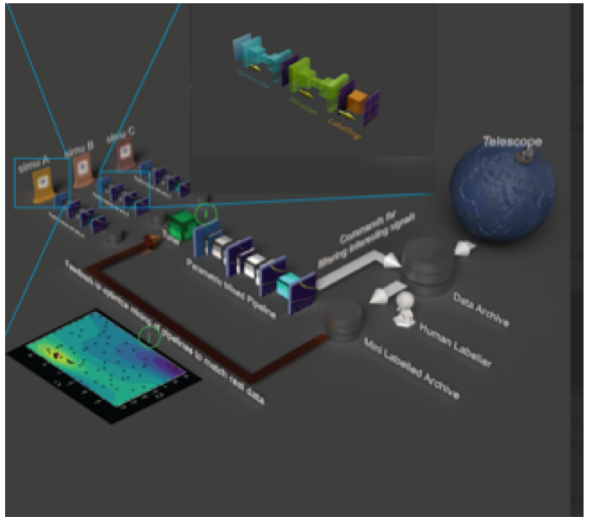


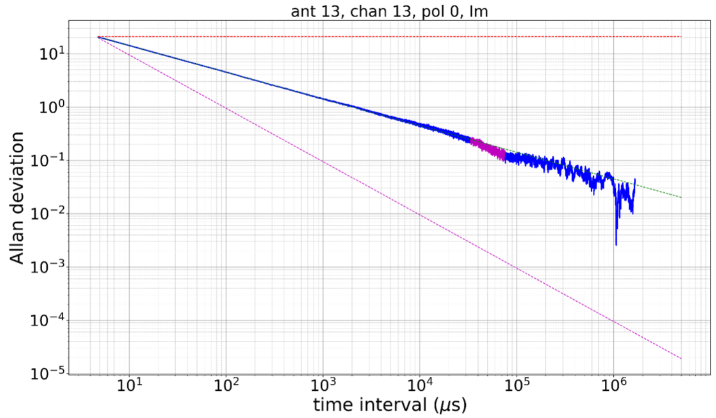
[htttps://gitlab.dzastro.de/punch/ml-ppa]

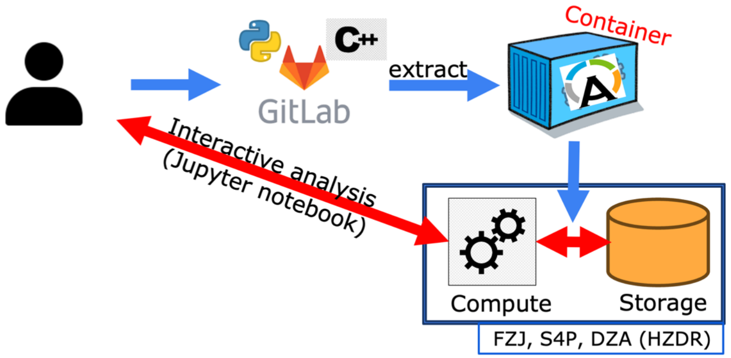


ML-PPA - Components

- Simulation + Machine Learning
  - Software Framework for Pulsar Detection using Machine Learning and Digital Twins [Tanumoy Saha]
  - CNN-based model [Andrei Kazantsev]
- Electronic noise
  - Effelsberg, MeerKAT: white noise [Yurii Pydopryhora]
  - Modular telescope simulator [Yannik Yiannakis]
- Software Engineering [Marcel Trattner]
  - CI/CD pipeline, automated testing, automated documentation
  - Profiling for transforming Python methods to C++
- Cloud computing
  - Interactive use of containers via Jupyter notebooks [Marcel Trattner]
  - Integration into HPC @ DZA [S. Borra, U. Canbolat, M. Drobek, L. Haupt]

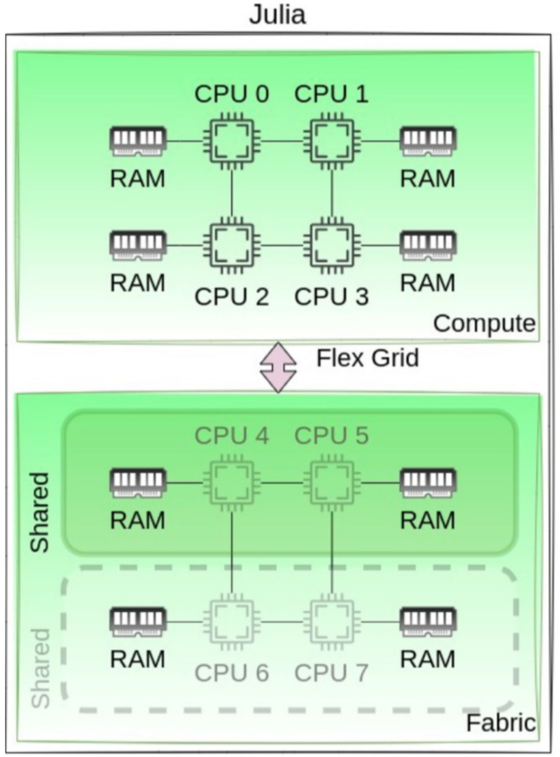




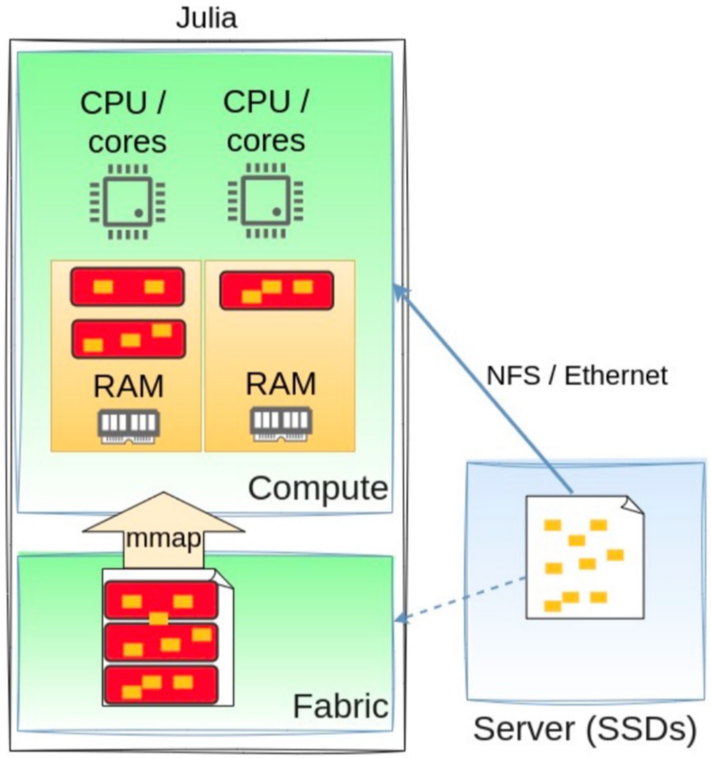


Memory-based computing – first results (Elsa Buchholz)

Memory-based Computing – First Results [Elsa Buchholz]



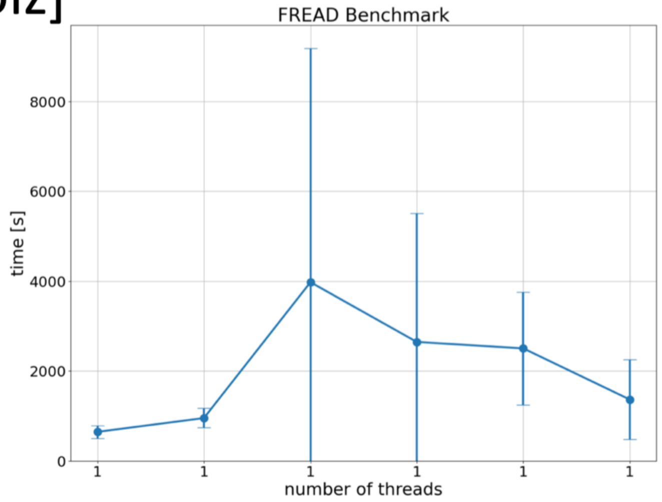
- 64 CPUs, 28 cores / CPU
- 48 TB RAM
- Flex Grid: 12 GB/s
- CPUs split between “compute” and “fabric” (~ super-fast disks)

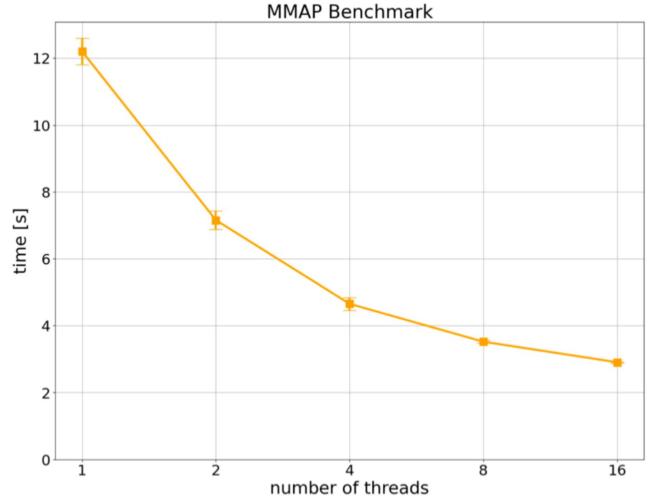


Simplistic SQL-like workflow: analysing 1 million randomly distributed small chunks (1 kB) from 1.5 TB file:

(a) file mounted over NFS/Ethernet

(b) file in-memory (via mmap from fabric)





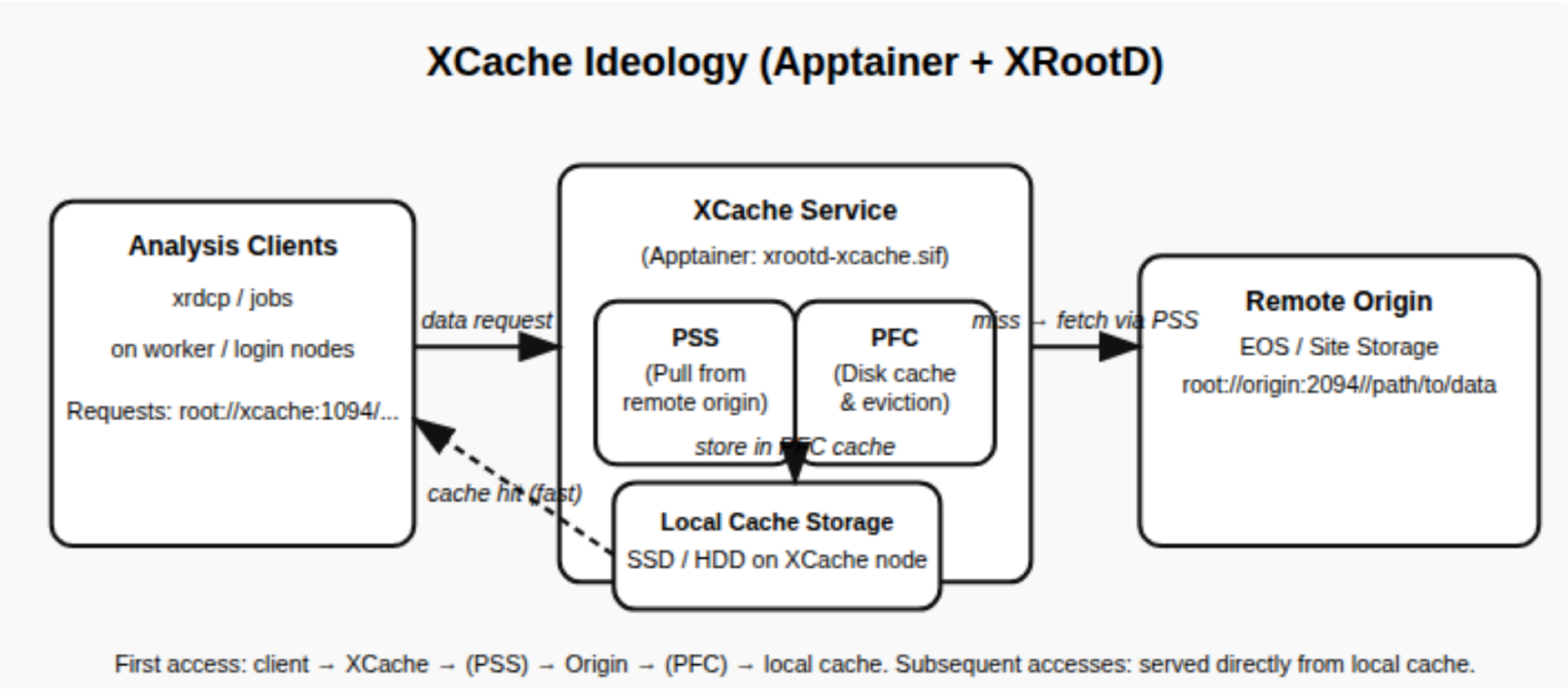
In-memory workflow (b) ~ 100 times faster than conventional workflow (a)

Efficient caching

Self-contained XRootD XCache Container

- Builds XRootD v5.6.4 inside Apptainer image
- Configured as a proxy cache on port 1094
- Upstream: eospublic.cern.ch:1094
- Persistent cache stored at /var/cache/xrootd

Minimal xcache.cfg enables the Proxy Storage System (PSS) for remote data access and the Persistent File Cache (PFC) for **local caching**, creating a **simple and efficient XRootD cache**.

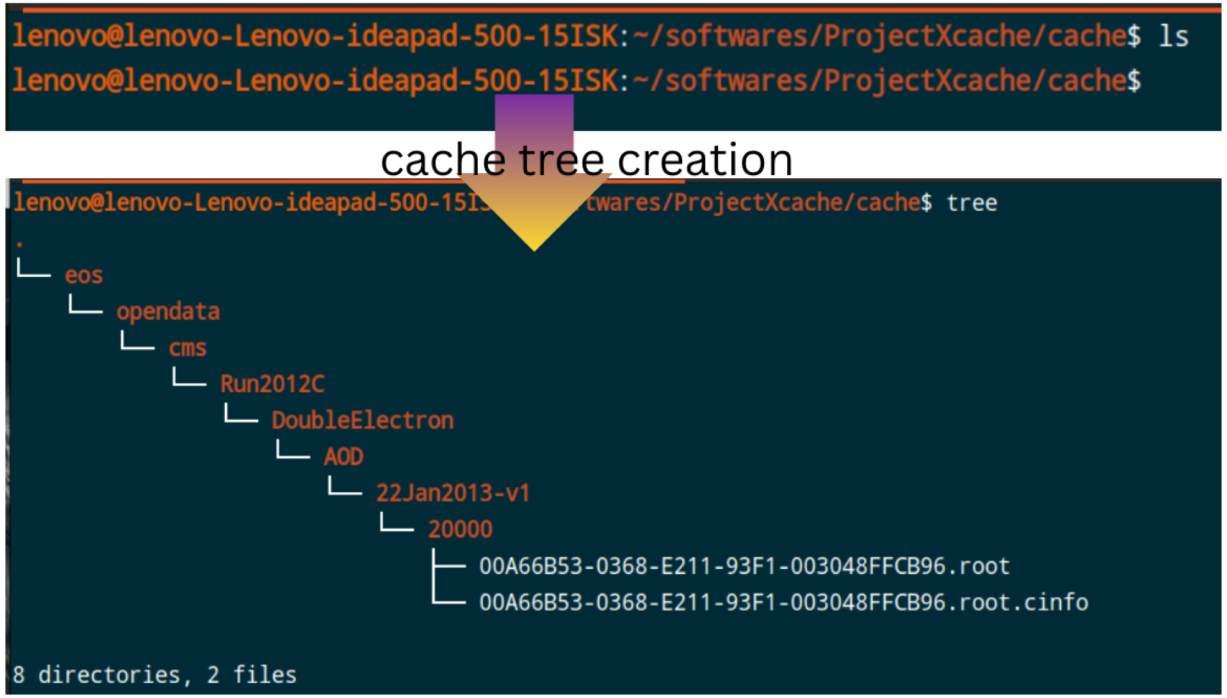


Efficient caching

Self-contained XRootD XCache Container

GIT repo
https://github.com/GautamDange/Xcache

RUN
apptainer exec \
--bind /tmp:/tmp \
xrootd-xcache.sif \
xrootd -c /etc/xrootd/xcache.cfg -l - -f



TEST
time xrdcp
root://$XCACHE_ADDR//eos/opendata/cms/Run2012C/
DoubleElectron/AOD/22Jan2013-v1/20000/00A66B53-
0368-E211-93F1-003048FFCB96.root .

Efficient caching

Self-contained XRootD XCache Container

Speed up calculation
•First download (no cache) took ≈ 891.7 s.
•Second download (via XCache) took 25.152 s.
XCache made this transfer about 35× faster than the direct download.

Currently it is super high as cache is on same system as of client.
It will vary with file size and also when cache server will be a local cache machine on LAN.

→ Under more systematic investigation

```
lenovo@lenovo-Lenovo-ideapad-500-15ISK:~/softwares/xrootdtest$ export XCACHE_HOST=10.83.15.16
lenovo@lenovo-Lenovo-ideapad-500-15ISK:~/softwares/xrootdtest$ export XCACHE_PORT=1094
lenovo@lenovo-Lenovo-ideapad-500-15ISK:~/softwares/xrootdtest$ export XCACHE_ADDR=$XCACHE_HOST:$XCACHE_PORT
lenovo@lenovo-Lenovo-ideapad-500-15ISK:~/softwares/xrootdtest$ time xrdcp root://$XCACHE_ADDR//eos/opendata/
cms/Run2012C/DoubleElectron/AOD/22Jan2013-v1/20000/00A66B53-0368-E211-93F1-003048FFCB96.root .
[3.862GB/3.862GB][100%][==================================================][4.438MB/s]

real    14m51.731s
user    0m3.892s
sys     0m11.201s
lenovo@lenovo-Lenovo-ideapad-500-15ISK:~/softwares/xrootdtest$ ls
00A66B53-0368-E211-93F1-003048FFCB96.root
lenovo@lenovo-Lenovo-ideapad-500-15ISK:~/softwares/xrootdtest$ rm *
lenovo@lenovo-Lenovo-ideapad-500-15ISK:~/softwares/xrootdtest$ time xrdcp root://$XCACHE_ADDR//eos/opendata/
cms/Run2012C/DoubleElectron/AOD/22Jan2013-v1/20000/00A66B53-0368-E211-93F1-003048FFCB96.root .
[3.862GB/3.862GB][100%][==================================================][158.2MB/s]

real    0m25.152s
user    0m1.851s
sys     0m6.681s
```

Speed-up is about **35.5×**.

- First run (no cache): **14m 51.731s ≈ 891.731 s**
- Second run (with XCache): **0m 25.152s ≈ 25.152 s**

$$S = \frac{T_{\text{no cache}}}{T_{\text{cache}}} = \frac{891.731}{25.152} \approx 35.5$$

Update for WP5

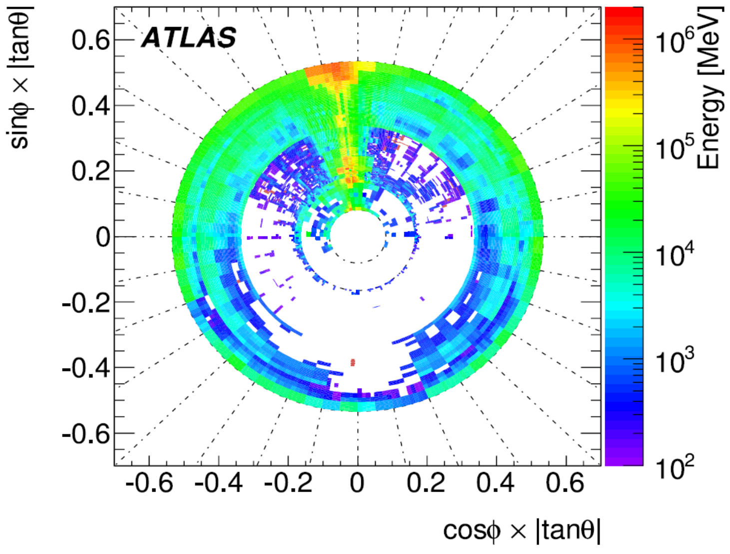
Development of ML prototypes for anomaly detection and predictive maintenance

Critical task for future online workflows

Potential for generalization

Study at TU Dresden

- Anomaly detection in real-time data processing of particle detectors:
  - Search for unusual signals → interesting physics or detector problems?
- Explorative study of detection of noise bursts in the ATLAS Liquid Argon Calorimeters
  - beam-related noise bursts induced by high-voltage cables - duration O(μs)
  - goal: **tag the noise data in real-time** to **exclude them from physics data stream**

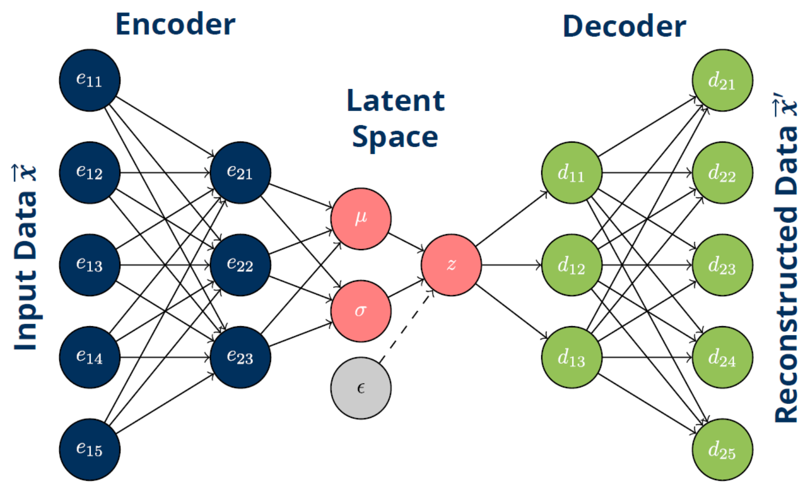


ATLAS Collaboration, CERN-PH-EP-2014-045

Coherent noise burst in transverse plane of ATLAS LAr Calorimeter faking a very large signal

Anomaly detection – noise bursts

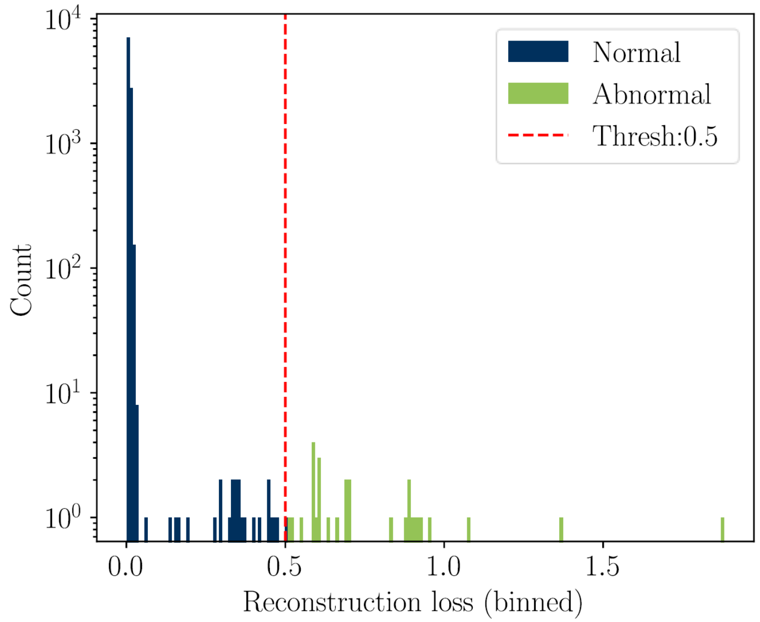
- Variational auto-encoder:
  - learn regular detector pulse sequences and condense information into a reduced "latent space"
  - find anomalies of signals with excessive reconstruction loss

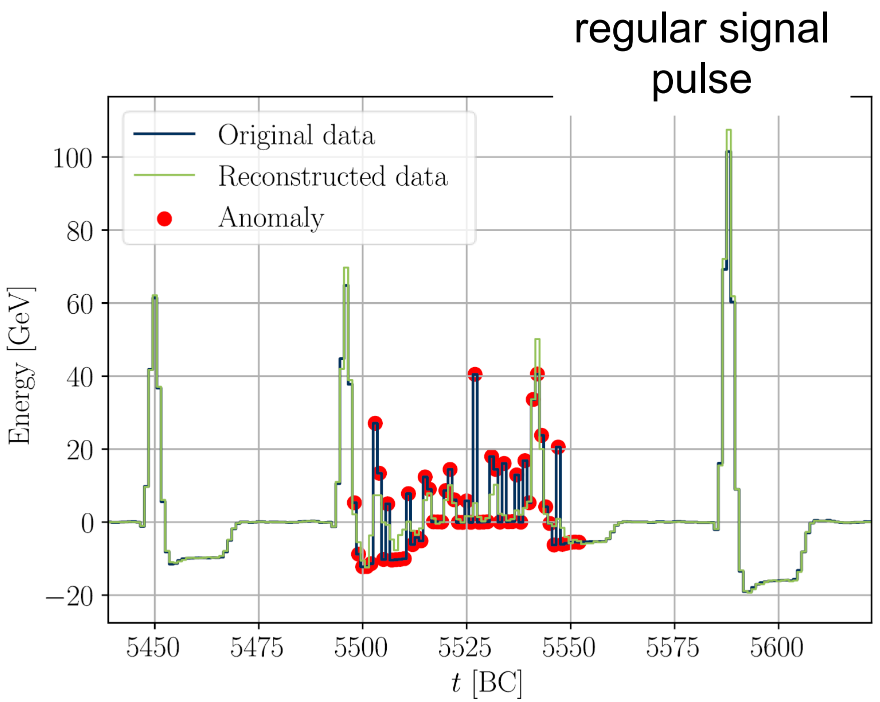


Loss function:

$$\mathcal{L} = (1-\beta)\mathcal{L}_{\text{Rec}}(\vec{x}',\vec{x}) + \beta \cdot D_{\text{KL}}(\mathcal{N}(\mu,\sigma)|\mathcal{N}(0,1))$$

$$\boxed{\mathcal{L} = (1-\beta)\mathcal{L}_{\text{Rec}}(\vec{x}',\vec{x}) - \frac{\beta}{2}(\log(\sigma^2) + 1 - \mu^2 - \sigma^2)}$$

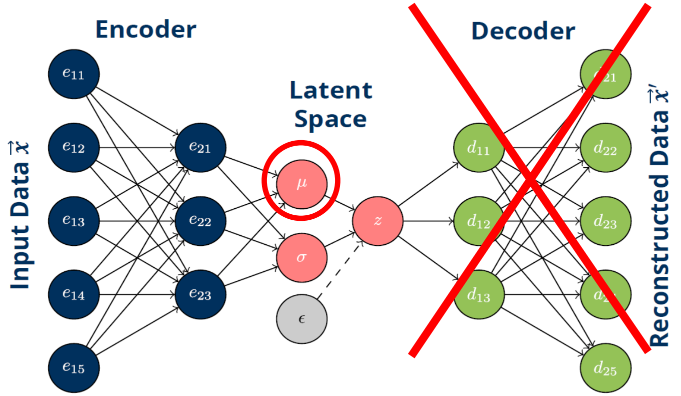


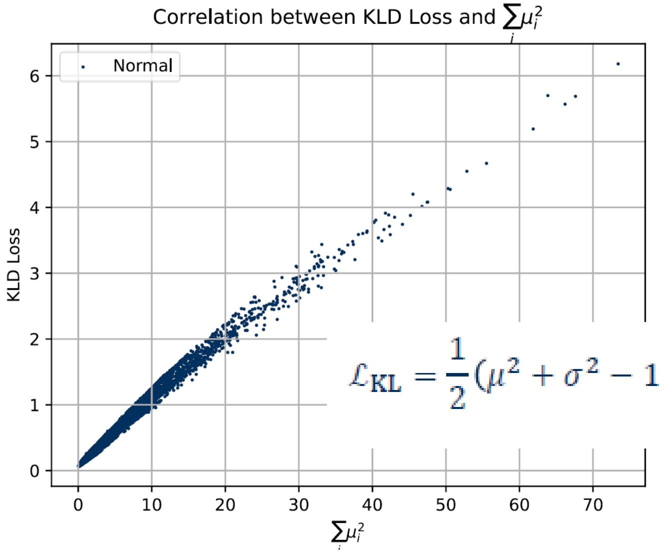


Anomaly detection – noise bursts

- Application shall be implemented in Field Programmable Gate Arrays (FPGA) in future:
  - reduce the network size → remove decoder and only exploit the information available in the latent space
  - Kullback-Leibler loss can be further simplified to the summed mean-squares of the latent space variables

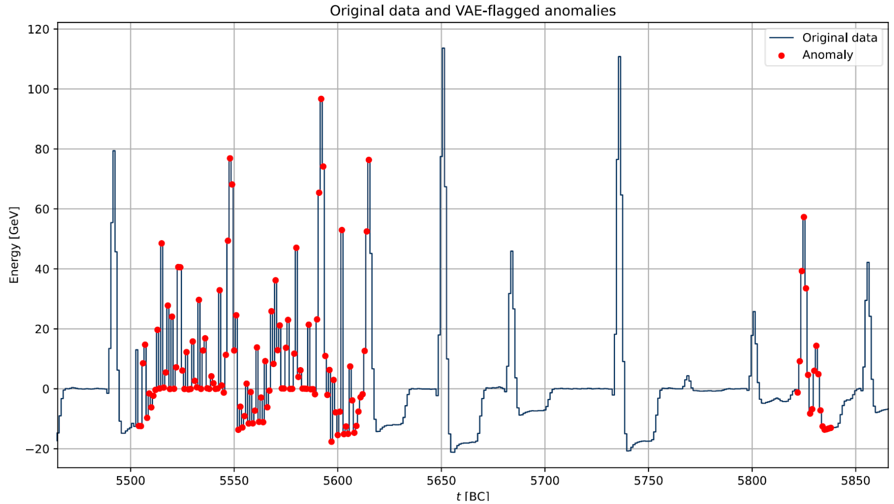
Parameter	Value
Input dim.	10
Dim. Layer 1	20
Dim. Layer 2	8
Latent dim.	6





$$\mathcal{L}_{\mathrm{KL}} = \frac{1}{2}(\mu^2 + \sigma^2 - 1 - \log(\sigma)) \quad \Rightarrow \quad \boxed{L = \sum_i \mu_i^2}$$

New score works for anomalies on only pileup data



- Anomaly tagging remains efficient also for reduced network size

- Noise detection relies on information from a single detector cell
- Problem: overlapping pulses are identified as "anomalous", too

→ include **cell-to-cell noise correlation to tag anomalies**

Anomaly detection – noise bursts

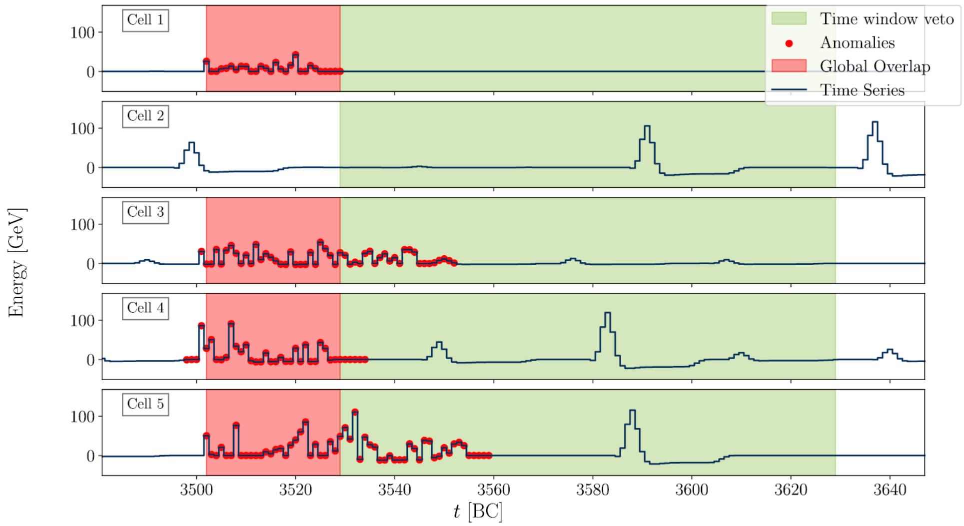
- Correlated noise tagging:
  - if a certain fraction of cells show anomalies
  → flag the data sequence and veto further treatment by trigger or data processing

- Next steps:
  - investigate more advanced methods to include information on noise correlation
  - quantify performance in comparison to off-detector noise tagging
  - implement prototype in FPGA firmware

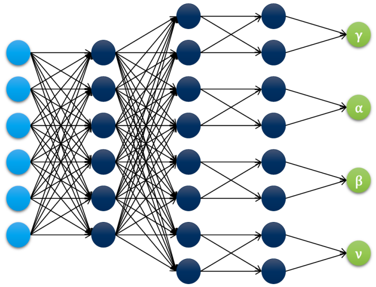
- Further developments:
  - estimate uncertainties of machine learning predictions by Deep Evidential Regression
  - additional network output nodes can quantify "epistemic" uncertainties
  - main challenge: keep the predictive power of original network, e.g. cell energy resolution of the ATLAS LAr Calorimeters, and obtain reasonable uncertainties

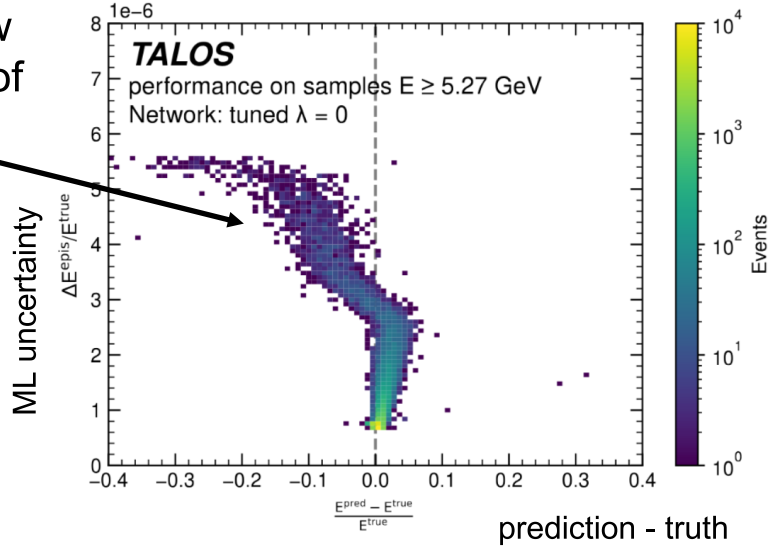
J. Herrmann, "Unsupervised Detection of Noise Bursts in Simulated LAr Signals with Autoencoders", TU Dresden

Multiple cell Anomaly Detection: Results



ML uncertainty does not yet show strong correlation with deviation of ML prediction from truth





prediction - truth

A. Wittwer, "Deep Evidential Regression for the Energy Reconstruction of the ATLAS LAr Calorimeters", TU Dresden

Outlook

TA5 workshop in spring 2026 under discussion

Public availability of “raw data” challenging

Progress in several areas related to time-critical workflows

Discussions about options of including TA5 developments in PUNCH results

Future Observations use case will follow up on parts of TA5 developments

Strong focus on remaining deliverables

BACKUP

- **TA5-WP1 Implications for discovery potential and reproducibility**
  - D-TA5-WP1-2 (Bielefeld) Report on impact of on-line filtering on FAIR principles
    → postponed to February
  - D-TA5-WP1-3 (Bielefeld) Concepts towards a general protocol on capturing the decisions made by, and status of, real-time sensors, as a basis for a future demonstator

- **TA5-WP2 Dynamic filtering**
  - D-TA5-WP2-3 (MPIfR) Test environment for identifying highly complex (multi-parametric) signals in huge data streams.
  - D-TA5-WP2-4 (Dresden, Mainz, MPIfR) Generic tools to both convert trained neural networks into efficient HLS/VHDL FPGA firmware optimised for a real-time, low-latency environment
    → availability of open data still in discussion
  - D-TA5-WP2-5 (FIAS, Dresden) Algorithms for massively parallel real-time sorting, clustering and pattern recognition on specialised hardware
  - D-TA5-WP2-6 Algorithms and Machine Learning methods for filtering and selecting relevant transient/anomalous signals
  - D-TA5-WP2-7 Pipeline for anomalous signal detection with low false-alarm probability for multi-messenger follow-up

- **TA5-WP3 Dynamic archiving**
  - D-TA5-WP3-2 (MPIfR) Present a framework in which queries to dynamic archives can be transformed into a dynamic filter (as used by some combination of sensors), and vice versa

    → Parts of the work done, to be deployed in framework of TA5-WP2-3 planned for Q4/25-Q1/26
  - D-TA5-WP3-3 (Humboldt-Uni) Present methods by which queries to dynamic archives also return an estimate on the potential of information loss, i.e. how well the archive response can be assumed to approximate the response of a real-time sensor

    → Parts of the work done in the context of AMPEL framework (paper currently written)

- **TA5-WP4 Scaling workflows**
  - D-TA5-WP4-3 (FIAS) Caching strategies for processing a set of benchmark files with the evaluated efficiencies and latencies

    → Development and setup done, stat. evaluation ongoing, to be finalized in 2025
  - D-TA5-WP4-4 (FIAS) Definition and initial implementation of an efficient real-time data processing framework
  - D-TA5-WP4-5 (HTW) Scaled feedback interfaces between off-line software (e.g. CASA) and selected real-time processes using MeerKAT data

TA5-WP5 Evaluation and validation of instrument response & characteristics

- D-TA5-WP5-1 (Dresden) Development of machine learning prototypes for anomaly detection and predictive maintenance
  → Successful tests using variational autoencoders for anomaly detection (for noise bursts)

- D-TA5-WP5-2 (MPIfR) Interference recognition and mitigation schemes for transient discovery leading to a robust triggering system

  → Part of work successfully completed in context of D-TA5-WP2-3, needs work to complete as standalone deliverable

- D-TA5-WP5-3 Expansion of the concept to a generalized toolkit for predictive maintenance and anomaly detection

- D-TA5-WP5-4 Evaluation of the machine learning approaches by analyzing false-alarm rates and online feedback